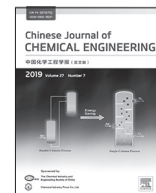




Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

journal homepage: www.elsevier.com/locate/CJChE

Review

Distributed control and optimization of process system networks: A review and perspective[☆]Wentao Tang, Prodromos Daoutidis^{*}

Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:

Received 12 June 2018

Received in revised form 26 July 2018

Accepted 7 August 2018

Available online 19 September 2018

Keywords:

Distributed control

Distributed optimization

Process networks

Decision making

ABSTRACT

Large-scale and complex process systems are essentially interconnected networks. The automated operation of such process networks requires the solution of control and optimization problems in a *distributed* manner. In this approach, the network is decomposed into several subsystems, each of which is under the supervision of a corresponding computing agent (controller, optimizer). The agents coordinate their control and optimization decisions based on information communication among them. In recent years, algorithms and methods for distributed control and optimization are undergoing rapid development. In this paper, we provide a comprehensive, up-to-date review with perspectives and discussions on possible future directions.

© 2018 The Chemical Industry and Engineering Society of China, and Chemical Industry Press Co., Ltd. All rights reserved.

1. Introduction

Large-scale and complex systems have become prevalent in chemical and energy industries. Such systems arise from the integration of process units and plants as a result of sustainability motivations, specifically, economic efficiency, energy recycle, and carbon and water footprint reduction [1,2]. Examples of such complex systems include chemical plants with recycles [3,4], crude oil refineries [5], smart grids and microgrids [6], integrated heating, ventilation and air conditioning (HVAC) systems [7], and supply chains [8,9].

Process control and optimization technologies are crucial to the implementation of such sustainable solutions in the chemical, energy and other relevant industry sectors. However, control and optimization become difficult for these large-scale and complex systems due to the size of their models and the existence of phenomena that emerge at a *network* level, such as multi-time-scale dynamics [10], multi-level uncertainties [11], and propagation of disturbances [12]. These challenges to the control and optimization of *process networks* call for theoretical efforts to deal with each individual phenomenon, and a computational architecture for control and optimization that accounts for their networked nature.

A promising approach to this end is to make the control and optimization decisions in a *distributed* manner. Specifically, a distributed architecture of control and optimization has the following meaning:

- The entire process system, as a network, is decomposed into several constituent subsystems.
- A computing agent (controller, optimizer) is associated with each subsystem and is responsible for making the decision in the corresponding subsystem.
- Some information is exchanged between the agents to coordinate their decisions towards the control or optimization objective of the entire network.

A distributed architecture is different from a centralized one, in the sense that the decisions are made by multiple agents rather than a single agent. Compared to its centralized counterpart, distributed control and optimization are more *scalable* and *parallelizable*, and hence can be more efficient computationally. Moreover, a distributed architecture is *flexible* to the entering, exiting, and changes of subsystems (“plug-and-play” property [13]), and can be also combined with distributed fault detection for a fault-tolerant control architecture [14,15]. It is also different from a decentralized architecture (which appears in traditional process control) due to the existence of some coordination mechanism, which generally enables the agents to account for other subsystems and arrive at desirable or optimal decisions for the overall network.

This paper provides a comprehensive and up-to-date review of distributed control and optimization. We aim to present the

[☆] Supported by Division of Chemical, Bioengineering, Environmental and Transport Systems (CBET) of the National Science Foundation (NSF) of the United States of America.

^{*} Corresponding author.

E-mail address: daout001@umn.edu (P. Daoutidis).

fundamental ideas and guide the readers through the most significant developments in recent years. The paper is organized as follows. We first introduce in Section 2 the preliminary concepts for the distributed architecture of control and optimization. Two major issues in developing a distributed control or optimization scheme, namely the methods of decomposing the network into subsystems, and the methods of performing optimization based on a network decomposition, are reviewed in Section 3 and Section 4, respectively. A case study on a reactor–separator process network is shown in Section 5. We discuss some important future directions in Section 6. Conclusions are made in Section 7.

2. Distributed Control and Optimization on Networks

2.1. Control and optimization

In the operations of process systems, many decisions need to be made. Decision making problems are organized in a hierarchy which contains the following problems on different time and space scales [8,16]:

- Control – input variables (flow rates, heat exchange rates, etc.) are manipulated to stabilize the system states (temperature, pressure, etc.) at their setpoints or track reference trajectories in a time scale of minutes to hours.
- Real-time optimization – the setpoints or reference trajectories are determined so as to optimize the economic performance under current market conditions. Such decisions are made in the time scale of hours to days.
- Scheduling – the sequence, timing and processing quantities are determined for the production equipment to meet the production goals over a time scale of days to weeks.
- Planning – high-level decisions such as production levels and product inventories are determined based on demand and marketing forecasts over weeks to months.
- Supply chain management – decisions on the material, information and financial flows between different business entities, including suppliers, producers, distributors and retailers, are made for enterprise-wide optimization.

An illustration of such a hierarchy is shown in Fig. 1. Traditionally, these problems are solved separately, where any lower layer decisions are made under given higher layer decisions. The integration of these decision making problems, which is necessary especially under dynamic market conditions, is an area of active recent research [17–20].

The aforementioned operational decisions are typically determined through the solution of the corresponding optimization problems. Such optimization formulations are well established in the literature of process systems engineering [21,22] by specifying the relevant variables, defining the objective function, and modeling the process system as the constraints involving these optimization variables. For example, using a state-task or resource-task network to model the process units and materials, mixed-integer linear and nonlinear programming (MILP, MINLP) formulations can be derived for scheduling problems (see e.g. [23–25] for reviews). Optimization is also applied to the control decisions in the form of model predictive control (MPC) [26,27]. If the objective function is an economic one, the control strategy is called economic model predictive control (EMPC) [28,29]. In these approaches, suppose that we have the dynamical model of a process system:

$$\dot{x} = f(x(t), u(t)) \quad (1)$$

where $x(t)$ and $u(t)$ are the vectors of states and control inputs at time t , respectively, and f is a vector-valued function. At any time t , the inputs are determined by solving the following optimal

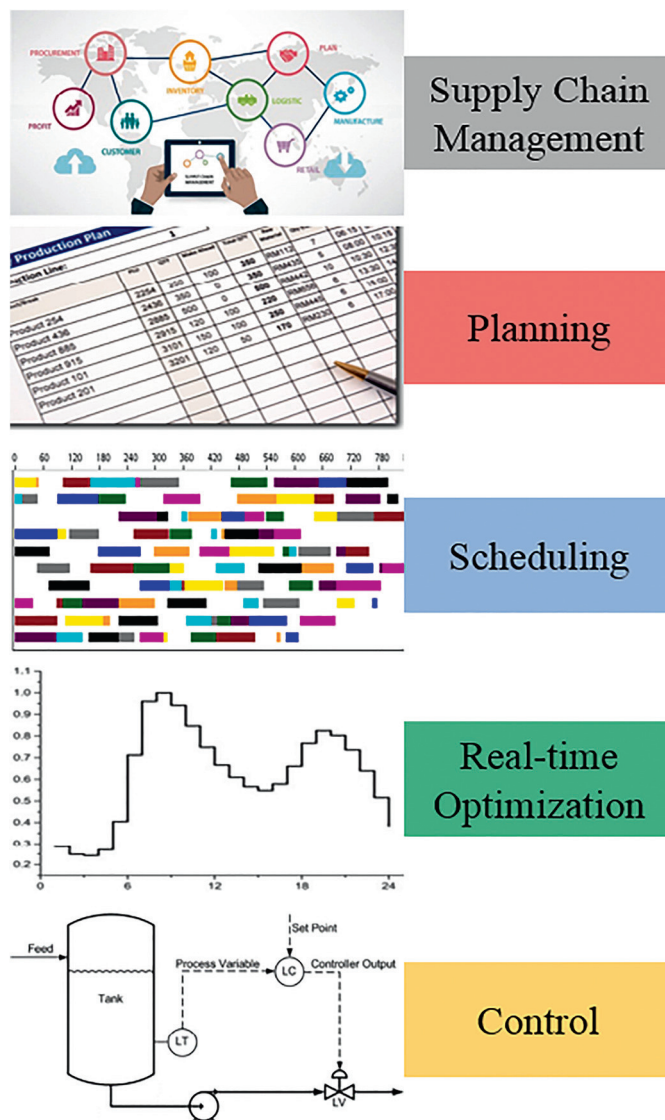


Fig. 1. The hierarchy of decision making in process systems operations.

control problem involving the inputs $\hat{u}(\tau)$ and predicted states $\hat{x}(\tau)$ in a future time horizon $\tau \in [t, t+T]$:

$$\begin{aligned} \min_{\hat{u}(\cdot)} \int_t^{t+T} \phi(\hat{x}(\tau)\hat{u}(\tau))d\tau + \psi(\hat{x}(t+T)) \\ \text{s.t. } \dot{\hat{x}}(\tau) = f(\hat{x}(\tau), \hat{u}(\tau)), \tau \in [t, t+T] \\ \hat{x}(\tau) = x(t) \text{ given.} \end{aligned} \quad (2)$$

Here the objective function is the integral of ϕ , which represents a cost or performance loss, and a terminal cost ψ . The constraints contain the system dynamics, and possibly additional operational constraints such as input bounds $\hat{u}(\tau) \in U, \forall \tau \in [t, t+T]$. The problem (2) can be solved with different approaches through discretization of $u(\tau)$ and $x(\tau)$ in $[t, t+T]$. Then the control decisions at the current time t are determined as $u(t) = \hat{u}(t)$.

The centralized framework of decision making is represented in Fig. 2(a). The agent is the decision maker, namely a controller (optimizer), which receives the information required for decision making from the system and sends the decisions back to the system. For example, in control problems, the agent (controller) receives the state variables $x(t)$ from the sensors, determines the control inputs $u(t)$ (depending on $x(t)$ as a parameter, thus also denoted by $u(x(t))$) and sends them to the actuators to execute the control

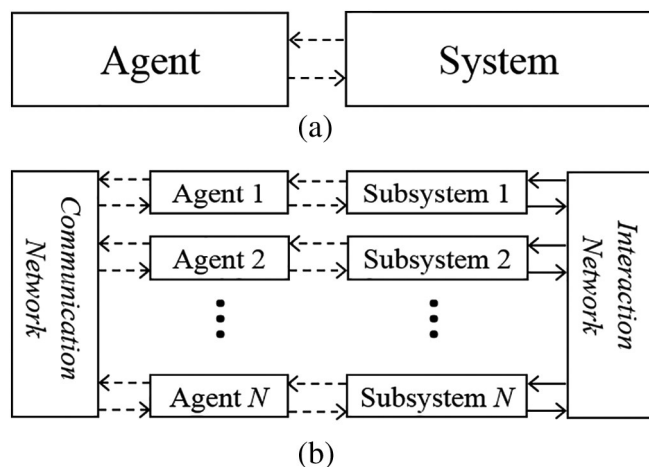


Fig. 2. Decision making in (a) centralized and (b) distributed architectures.

signals. The agent and system thus compose a closed-loop system $\dot{x} = f(x(t), u(x(t)))$ evolving with time.

2.2. Distributed control and optimization

For large-scale and complex systems, centralized decision making using the monolithic system model is usually computationally difficult, since in most cases, the associated optimization problems are intrinsically nonscalable as the problem size grows. Also, due to the dependence of all the decisions on all the system information, such a centralized control or optimization scheme is inflexible to any changes or faults in the plant, actuators or sensors. To avoid such limitations, distributed computing offers a more scalable, parallelizable and flexible paradigm for decision making [30]. In recent extensive research works, this has been addressed in the contexts of distributed linear feedback control [31,32], distributed dissipativity-based control [33,34], distributed nonlinear model predictive control [35–37], distributed power scheduling [38,39] and economic real-time optimization [40,41].

The architecture for such a decision making is shown in Fig. 2(b). The entire network is decomposed into constituent subsystems with interactions between them, e.g., a tightly integrated chemical plant is partitioned into groups of process units. An agent is associated with each subsystem. The agents receive information from the corresponding subsystems, generate decisions, and transmit the determined decisions back to the subsystems. Due to the existence of interactions among the subsystems, it is insufficient to make decisions with separate agents, i.e. to use a decentralized architecture where each agent makes part of the decisions on its own. Instead, communication channels need to be established among the interacting agents, through which the agents share certain information such as their local measurements and decisions. In this way, the mutual influences between the subsystems can be accounted for by the agents, and hence the computation of all the agents can be coordinated to meet an objective for the entire system.

In the context of distributed control and optimization, we consider the following aspects to be important:

- The development of network decomposition methods and the studies on the relation between decompositions and the resulting performance help to specify an optimal decomposition for distributed control and optimization.
- The development of distributed optimization algorithms and the studies on their convergence properties and numerical performance help to implement a more computationally efficient decision-making scheme.

- The non-ideality of communications will induce constraints on the control and optimization performances. The methods of analyzing and accommodating these communication network-induced issues help to improve the feasibility and flexibility of distributed decision.

Among the above three aspects, the communication issues have been addressed in a less unifying and non-universally applicable way. We will review the network decomposition methods and distributed optimization methods in the following two sections, respectively, and communication issues in the following subsection.

2.3. Issues with communication network

In distributed control and optimization, the communication among the agents as well as the communication between the agents and the system, are enabled by a communication network technology, including field-bus, controller-area network (CAN), Ethernet and Internet [42,43]. However, any communication network has a limited capacity, which results in several so-called network-induced constraints [44]. The influence of such network-induced constraints on the distributed optimization or closed-loop performance needs to be analyzed or accommodated.

In the context of distributed optimization using communication networks, the following problems have been studied:

- When the agents have different processing speeds, asynchronous iterations can be computationally more efficient than the ideally synchronized iterations. Asynchronous algorithms of distributed optimization have recently been developed [45,46].
- Due to the quantization of signals transmitted between the agents, distributed optimization algorithms with inexact communication have been developed [47,48], and the conditions on quantization for maintaining convergence properties have been studied [49].
- Event-triggered communication can reduce the communication cost, where the agents maintain estimations of the decisions of its neighboring agents, and the communication is triggered when this estimation deviates from the true values beyond a threshold [50,51]. Such methods have been applied to optimal power flow problems [52,53].
- Another approach to reduce the communication requirement is to develop communication-efficient distributed optimization algorithms [54] with a tradeoff between exact steps with communication and inexact steps with only local information [55–57].

The effect and accommodation of network-induced constraints on the closed-loop system have been extensively studied in the context of networked control systems, assuming that an explicit form of control law is given, or that the system and the controller are prescribed to be linear. Typically, these constraints are regarded as uncertainties or combined with the original dynamics into a stochastic switched system or hybrid discrete–continuous system. Based on such models, the existence and synthesis of Lyapunov functions are used for stability analysis. These network-induced constraints include:

- Time delay on the network. Time delays were treated as uncertainties in a robust stability framework [58] or in the context of switched systems [59]. Delays can also be considered in the stability analysis of model predictive control [60–62].
- Missing and time reversal of transmitted information, i.e. package dropout and disorder. These can be accommodated by robust controller design [63], adaptive control [64,65], or transmitted data checking [66]. Simultaneous package dropout and time delay were considered in [67].

fine-tuning procedure that seeks to minimize the cut edges (see e.g. [89]). We note that for distributed computing, it is desirable to impose balanced computational loads on the agents, which requires the sizes of the resulting subsystems to be uniform. To this end, approximately equal-size partitioning algorithms have also been proposed in [90] and applied to the Barcelona water network. We note that graph partitioning algorithms have wide applications in computer science and very-large-scale integration (VLSI) technology. For recent developments in graph partitioning, the readers are referred to [91].

3.2. Clustering algorithms

Clustering algorithms are widely used in the areas of data mining and machine learning. By defining a distance metric between the data points, clusters (groups) of mutually close points are formed, which helps to reveal the underlying organization of the dataset [92,93]. Hierarchical clustering is a major class of clustering algorithms [94], which generates a dendrogram of clusterings through successive agglomeration or division.

Hierarchical clustering of input and output variables was used to decompose control systems based on relative degrees [95] as input–output distance measures. In [96], the inputs and outputs are first paired with minimized total relative degrees, and the input–output pairs are then agglomeratively clustered based on their mutual distances. An improved implementation of this algorithm was proposed in [97], and an extension to control systems with both ordinary and partial differential equations was addressed in [98]. In [99], divisive hierarchical clustering method is used, where the dendrogram is generated through recursive bisections, and each bisection is determined by properly defined quality indices.

For illustration, Fig. 5 illustrates the pairing of input and output variables and the dendrogram obtained by hierarchical clustering of the input–output pairs for a solid oxide fuel cell system [96]. We note that the hierarchical clustering algorithm usually allows multiple solutions without determining an optimal decomposition.

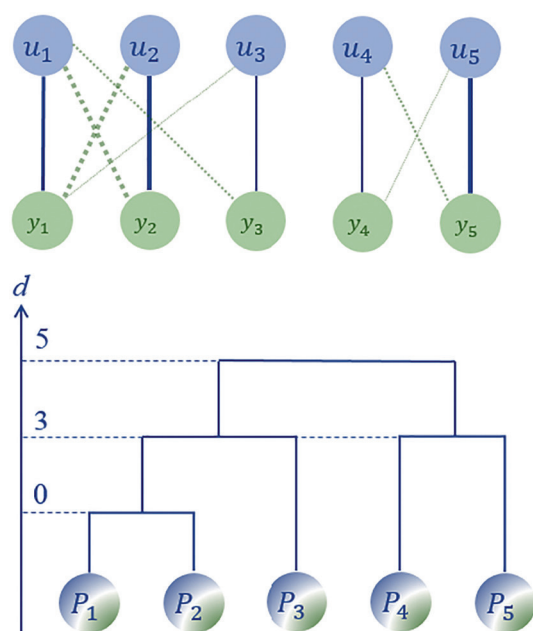


Fig. 5. Pairing and hierarchical clustering of inputs and outputs generated for a solid oxide fuel cell system. u and y represent inputs and outputs, respectively, and $P_1, \dots, P_5$ are input–output pairs. d is an input–output distance measure.

3.3. Community detection algorithms

Different from the approaches in the previous subsections, community detection as a major area of study in network science [100], provides a new perspective and efficient tools for decomposing large and complex networks.

Network science focuses on the macroscopic, statistical and coarse-grained properties of networks. By investigating examples of biological and social networks [101], it has been recognized that in many networks, the nodes can be assigned to several groups, called communities, such that the interactions inside the communities are stronger (denser) than the interactions between communities with a statistical significance. The mainstream community detection algorithms, including the well-known Louvain algorithm (fast unfolding algorithm) [102], belong to the type of modularity maximization [103,104], where a quantitative index called modularity is defined for any grouping of the nodes, and the grouping with approximately maximum modularity is found.

For example, the modularity-based community detection algorithm for unipartite undirected networks [103] is described as follows. Suppose the network is represented by an $n \times n$ adjacency matrix $A = [a_{ij}]$ where $a_{ij} = 1$ if there is an edge between node i and node j , and $a_{ij} = 0$ if nodes i and j are non-adjacent, $i, j = 1, \dots, n$. Let k_i be the degree of node i , i.e. the number of edges incident to node i , and $2m$ the total number of edges (with each edge double counted):

$$k_i = \sum_{j=1}^n a_{ij}, \quad 2m = \sum_{i=1}^n k_i = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \quad (3)$$

The modularity of any grouping of the nodes is defined as

$$Q(C_1, \dots, C_n) = \sum_{i=1}^n \sum_{j=1}^n \left(\frac{a_{ij}}{2m} - \frac{k_i k_j}{4m^2} \right) \delta(C_i, C_j) \quad (4)$$

where C_i is the community affiliation of node i , and $\delta(C_i, C_j)$ is 1 if $C_i = C_j$ and 0 otherwise. In the parentheses, $a_{ij}/2m$ is the fraction of edges lying between nodes i and j ; $k_i k_j / 4m^2$ is the fraction of edges incident to i , and thus $k_i k_j / 4m^2$ is the expected fraction of edges lying between i and j when the edges in the network are randomly redistributed while keeping the degrees of all the nodes constant. Therefore the difference between $a_{ij}/2m$ and $k_i k_j / 4m^2$ captures the statistical significance of the connectedness of nodes i and j . Q defined in Eq. (4) is the intra-community sum of such differences, and characterizes the statistical significance of the existence of communities as specified by $(C_1, \dots, C_n)$. The modularity-based community detection is thus an (approximate) maximization of Q over $(C_1, \dots, C_n)$.

Many other efficient algorithms have also been developed for detecting the community structures in networks (see [105] for a review), including the following representative ones. In label propagation (LPA) [106] and speaker-listener LPA (SLPA) algorithms [107], the nodes hold labels which keep being propagated between the adjacent nodes, and the converged labels are considered as the community affiliations. In Infomap [108], random walks are associated with the network to proxy the information flows on it, based on which the optimal information compression reveals the community structures. Stochastic block model (SBM) methods, which consider the community affiliations as latent labels of the nodes and infer these labels from the network information, have been a research focus more recently [109,110].

By decomposing a network into its communities, one can determine subsystems with strong intra-subsystem interactions and weak inter-subsystem interactions. Such a decomposition optimally concentrates the interactions inside the subsystems in the sense of statistical significance, and thus allows a distributed

control or optimization architecture with minimal compromise in performance.

For decomposition of control systems, several network representations have been developed [111–113]. Specifically,

- in [111], a directed graph representation called system digraph (see also *e.g.* [86]), is used to capture the structure of the interactions between input, state and output variables;
- in [112], the edges in the directed graph are weighted using sensitivities (coefficients in the linearized system, see also *e.g.* [114]), which capture the short-time dynamic response intensities, and a bipartite graph of inputs and outputs is constructed based on shortest path search in the directed graph;
- in [113], the edges in the input–output bipartite graph are weighted based on the relative time-averaged gain array, which is a measure of input–output interactions on a finite time scale (which typically matches the closed-loop time constant).

Based on the network representations, suitable community detection algorithms are applied correspondingly to generate the network decomposition for distributed control. As an illustration, Fig. 6 shows the community-based decomposition of a vinyl acetate monomer plant, which involves 26 manipulated inputs, 43 measured outputs, and 246 state variables (the detailed model is given in [115]).

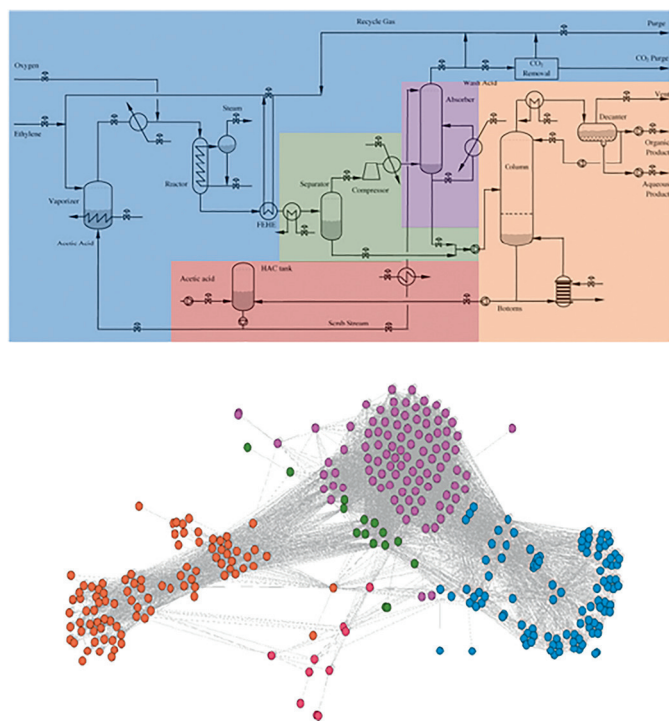


Fig. 6. Decomposition of vinyl acetate monomer plant (upper) based on community detection (lower).

For decomposition of optimization problems, community detection has also been proposed as a powerful method, and is currently undergoing active research. In [116], a unipartite graph representation of optimization problems was proposed and its decomposition by community detection was used to solve design optimization problems. In [117], a variable–constraint bipartite network and unipartite network representations of optimization problems were proposed, and the benefit of community detection-based decompositions was demonstrated with an artificial convex problem and an optimal control problem solved using different distributed

optimization algorithms. In [118], some mixed-integer nonlinear programming problems that are difficult to tackle with current solvers were investigated, and it was found that the decomposition by community detection helps to shrink the duality gap with tighter upper or lower bounds.

4. Methods of Decomposition-based Optimization

In the previous section we have presented methods of network decomposition. Once a decomposition is determined for the process, one can solve large-scale optimization problems by employing distributed optimization algorithms, which prescribe how the distributed agents update their decisions and how the information is transferred among the agents. In this section, we introduce the basics of the two major classes of formulations and algorithms of performing distributed optimization, namely block coordinate descent (BCD) algorithms and Lagrangian/augmented Lagrangian algorithms. We also review recent developments in optimization algorithms and software implementations in view of their importance in improving the computational efficiency of distributed optimization.

4.1. Block coordinate descent (BCD) algorithms

We consider optimization problems where the objective function involves N groups of decision variables $x_1, \dots, x_N$, and the constraints are separable, *i.e.* the constraints only exist within each variable group:

$$\begin{aligned} \min & f(x_1, x_2, \dots, x_N) \\ \text{s.t. } & x_i \in X_i, \quad i = 1, \dots, N. \end{aligned} \quad (5)$$

The separability of the constraints indicates that the decisions can be made separately without causing infeasibility. On the other hand, the variables are entangled in the objective function f (otherwise the problem (5) is separated). Therefore, whenever a group of decision variables is updated, the information of other variables and the overall objective function is needed. In other words, each computing agent needs to communicate the other agents with their decision variables, and keep the knowledge of the entire system on how all the variables affect the objective function.

Under such an architecture, the problem (5) can be solved by a BCD algorithm [119], where each group of decision variables (*i.e.* block of coordinates) is optimized with all the other variables fixed at their current values, and the variable updates are iterated until convergence. Namely,

$$\begin{aligned} x_1 &:= \arg \min_{x_1 \in X_1} f(x_1, x_2, \dots, x_N) \\ x_2 &:= \arg \min_{x_2 \in X_2} f(x_1, x_2, \dots, x_N) \\ &\dots \\ x_N &:= \arg \min_{x_N \in X_N} f(x_1, x_2, \dots, x_N) \end{aligned} \quad (6)$$

where $:=$ stands for value assignment, and $\arg \min$ is the argument of minimization, *i.e.* the minimizer. Fig. 7 illustrates the distributed optimization architecture with a BCD algorithm to compute the decisions with 4 agents.

This architecture is widely employed in distributed model predictive control [120–122] and economic model predictive control [123,124]. The control inputs in the prediction horizon for the N subsystems $u_1, \dots, u_N$ are considered as the decision variables. The decision variables are subject to separable constraints ($u_i \in U_i$, $i = 1, \dots, N$), but entangled in the objective function $J(u_1, \dots, u_N)$, which, for any given inputs $u_1, \dots, u_N$, is evaluated by extracting information from the corresponding trajectories predicted by an ordinary differential equation simulator (*e.g.* using the Runge–Kutta method). With such a formulation, each agent needs to

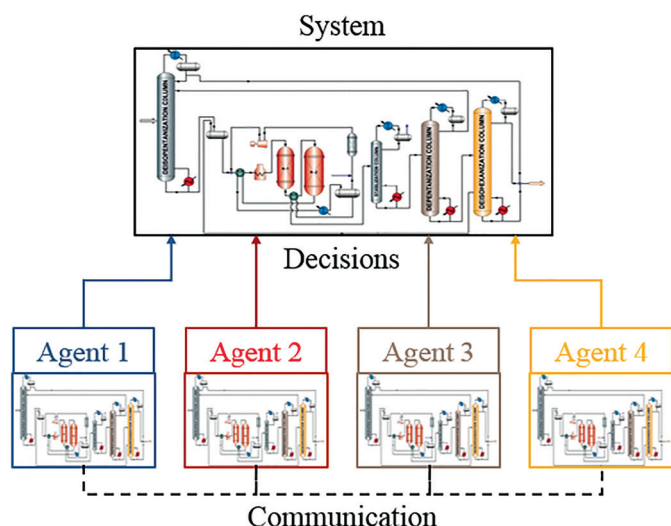


Fig. 7. Distributed optimization architecture with BCD algorithm. The agents have the knowledge of the entire system, and communicate their decisions mutually.

communicate its decisions with all other agents and performs optimization based on the entire system model, i.e.

$$\begin{aligned} u_1 &:= \arg \min_{u_1 \in U_1} f(u_1, u_2, \dots, u_N) \\ u_2 &:= \arg \min_{u_2 \in U_2} f(u_1, u_2, \dots, u_N) \\ &\vdots \\ u_N &:= \arg \min_{u_N \in U_N} f(u_1, u_2, \dots, u_N) \end{aligned} \quad (7)$$

The pristine algorithm in Eq. (6) has many variants. For example, instead of updating variables cyclically with fixed order, one may choose different selection rules of variable blocks, e.g. a randomized update rule [125], or parallelized update procedures [126]; also, instead of directly minimizing f in each iteration, one may approximate f with a local linearization with proximal terms (e.g. quadratic regulation) [127]. Under all these modifications, BCD-type algorithms can achieve a sublinear convergence rate [128,129]. By incorporating Nesterov's acceleration technique [130], computationally efficient implementations have been developed for convex problems [131,132]. For nonconvex problems in general, BCD guarantees convergence under an assumption of the Kurdyka–Łojasiewicz property on the objective function f [133].

For distributed model predictive control, two major variants of BCD are typically considered, where the agents perform optimization in sequence and in parallel respectively [134]. It was found for a benzene alkylation plant that the iteration number needed for BCD to solve each optimal control problem is small (convergent in about 15 iterations) [134].

4.2. Lagrangian and augmented Lagrangian algorithms

We consider optimization problems where the objective function can be separated into N terms ($f(x) = f_1(x_1) + \dots + f_N(x_N)$), each involving a group of decision variables, and linear equality constraints may exist between different variable groups in addition to the constraints within these groups:

$$\begin{aligned} \min & \sum_{i=1}^N f_i(x_i) \\ \text{s.t.} & \sum_{i=1}^N A_i x_i = b \\ & x_i \in X_i, \quad i = 1, \dots, N. \end{aligned} \quad (8)$$

Since the variable groups are separable in the objective function, i.e. a change in x_i only influences the $f_i(x_i)$ term of the objective

function, each agent is allowed to keep the knowledge of only its subsystem model. On the other hand, due to the existence of linear constraints interconnecting different variable groups, any agent needs to communicate with its neighbors (the other agents whose corresponding subsystem contains variables interacting with the subsystem of this agent) to share the information of interacting variables. Fig. 8 shows such an architecture. Compared to the previous architecture in Fig. 7, this new distributed architecture is a truly *localized* and more flexible one, in the sense that only subsystem models and less communication burden between the agents are required.

With such an architecture, the problem (8) can be solved with the help of a *Lagrangian decomposition* [135]. Define the dual variables λ associated with the linear constraints (which have an economic interpretation of shadow prices), and the Lagrangian

$$L(x_1, x_2, \dots, x_N, \lambda) = \sum_{i=1}^N \bar{f}_i(x_i) + \lambda^T \left(\sum_{i=1}^N A_i x_i - b \right) \quad (9)$$

where the functions

$$\bar{f}_i(x_i) = \begin{cases} f_i(x_i), & x_i \in X_i \\ +\infty, & x_i \notin X_i \end{cases} \quad (10)$$

Thus the problem (8) is reformulated as

$$\max_{\lambda} \min_{x_1, \dots, x_N} L(x_1, \dots, x_N, \lambda) \quad (11)$$

The dual problem (11) is solved in a BCD-type algorithm, where the primal variables are minimized separately with the information of dual variables λ , followed by a dual update in its subgradient direction:

$$\begin{aligned} x_i &:= \arg \min_{x_i} L(x_1, \dots, x_N, \lambda) \\ &= \arg \min_{x_i \in X_i} [f_i(x_i) + \lambda^T A_i x_i], \quad i = 1, \dots, N \\ \lambda &:= \lambda + \alpha \left(\sum_{i=1}^N A_i x_i - b \right) \end{aligned} \quad (12)$$

The step size α for dual update is typically chosen by Fisher's rule [136], while other rules exist, such as the bundle method [137] and the surrogate subgradient method [138].

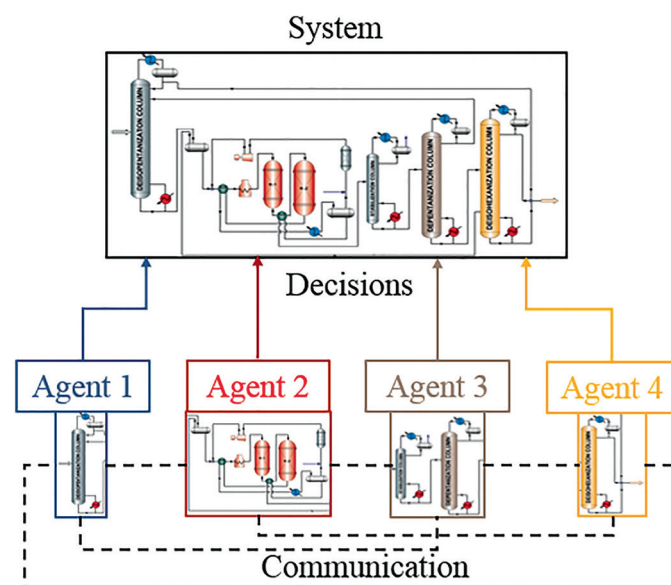


Fig. 8. Distributed optimization architecture with ADMM algorithm. The agents have the knowledge of the corresponding local subsystems, and communicate the interacting variables with their neighbors.

The performance of Lagrangian decomposition can be seriously affected by the nonconvexity of f_i and X_i . Alternatively, one can adopt an augmented Lagrangian, where a quadratic penalty term for the linear constraints with $\rho > 0$ is added to the Lagrangian to improve the convexity with respect to the primal variables [139]:

$$L(x_1, \dots, x_N, \lambda) = \sum_{i=1}^N f_i(x_i) + \lambda^T \left(\sum_{i=1}^N A_i x_i - b \right) + \frac{\rho}{2} \left\| \sum_{i=1}^N A_i x_i - b \right\|^2 \quad (13)$$

The so-called alternating direction method of multipliers (ADMM), as an augmented Lagrangian decomposition algorithm, is widely applied in machine learning [140]:

$$\begin{aligned} x_i &:= \arg \min_{x_i} L(x_1, \dots, x_N, \lambda) \\ &= \arg \min_{x_i \in X_i} \left[f_i(x_i) + \frac{\rho}{2} \left\| \sum_{i=1}^N A_i x_i - b + \frac{\lambda}{\rho} \right\|^2 \right], \quad i = 1, \dots, N \\ \lambda &:= \lambda + \gamma \rho \left(\sum_{i=1}^N A_i x_i - b \right) \end{aligned} \quad (14)$$

The step size parameter is chosen as $\gamma \in (0, (\sqrt{5} + 1)/2)$.

The application of Lagrangian decomposition to scheduling and planning problems has been discussed in many relevant works [22,141,142]. For distributed model predictive control, the application of Lagrangian decomposition [143–145] and ADMM [48,146,147] has also been proposed, for which the control decisions and the predicted states are taken as decision variables subject to constraints including the system dynamics. ADMM is also applied to other problems such as state estimation [148] and optimal power flow [149] on electricity grids.

Similar to BCD-type algorithms, variants of ADMM algorithms with parallelized [150] or randomized [151] variable updates have also been proposed, which are known to be sublinearly convergent for convex problems [152,153]. To accelerate ADMM for convex problems, Nesterov's technique has been proven to guarantee a quadratic convergence rate [154,155]. Alternatively, for faster convergence, one may optimally fix the penalty parameter ρ [156] or use an adaptive penalty parameter throughout the iterations [157]. For nonconvex problems, theoretical convergence under certain assumptions has also been discussed [158,159]. However, the acceleration of nonconvex ADMM seems more difficult. Recently, we proposed an accelerated nonconvex ADMM algorithm where a discounted Nesterov's acceleration scheme is incorporated into the parallelized ADMM [160].

4.3. Formulations, algorithms and software for optimization

In the previous subsections we have introduced BCD-type and Lagrangian-type distributed optimization algorithms and their corresponding communication architecture. Any distributed optimization algorithm is composed of iterated procedures of solving subproblems monolithically. It is desirable that these subproblems can be solved with high computational efficiency. To this end, we note that the following efforts are significant towards the efficient implementation of distributed optimization:

- Better *formulations* for problems of interest are being developed, which reduce the size or improve the convexity without losing the accuracy of the solution obtained. For example, to efficiently discretize continuous-time optimal control problems with small approximation error, collocation methods (pseudospectral methods) have been commonly adopted [161,162]; to efficiently solve complex scheduling problems, reformulation methods [163] and approximation refinement techniques [164] have been

proposed. The simplification of formulations using approximate or surrogate models seems especially crucial to integrated control and scheduling, which typically result in large-scale mixed integer dynamic optimization (MIDO) problems [165–167].

- Better *optimization algorithms* are also being developed to approach the optimum within fewer searching steps. For nonlinear programming problems (NLPs), e.g. for optimal control problems, the filtered line-search interior-point optimization (IPOPT) algorithm [168] usually exhibits a very fast convergence. Software packages have been developed to implement IPOPT as well as other interior-point optimization algorithms [169,170]. With IPOPT, the parametric sensitivity of nonlinear programs can also be computed [171]. Some recent works have focused on improving IPOPT by exploiting some special structures underlying the problem (e.g. optimal control problems typically have a tri-diagonal structure) [172,173]. For mixed integer nonlinear programs (MINLPs), a number of global optimization solvers – BARON [174], ANTIGONE [175], LINDO API [176], SCIP [177], COUENNE [178] – have been developed and compared on libraries of benchmark problems.
- Better *software platforms* are being developed to automate the formulation (modeling) of optimization problems with less user effort and implement the algorithms with less computational time. For the optimization modeling, different from the traditionally used environments such as GAMS and AMPL which directly formulate the problems, algebraic modeling languages such as Pyomo [179] and JuMP [180] have been recently proposed. These are based on high-level programming languages (Python and Julia, respectively) and enable object-oriented modeling (e.g. [181]), i.e. automatic instead of manual formulation. For the efficient evaluation of derivatives during the solution of optimization problems, automatic differentiation packages (e.g. CasADi) have been developed and interfaced with modeling languages and solvers [182,183].

The roles of modeling language, automatic differentiation and solver in the solution of optimization problems are illustrated in Fig. 9. So far, these efforts have been incorporated into toolboxes

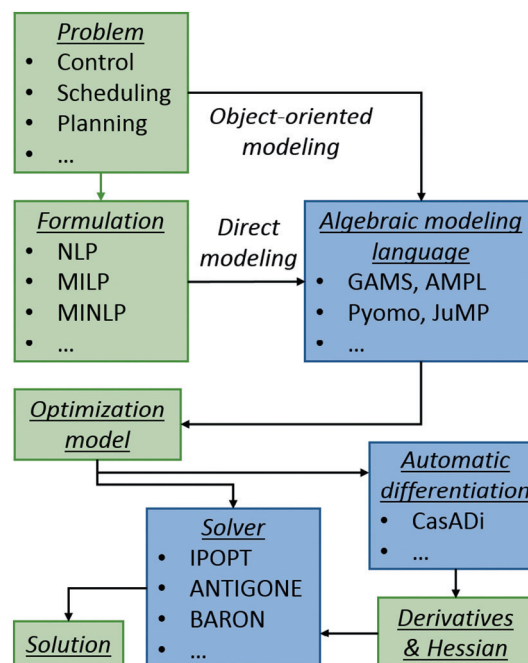


Fig. 9. The roles of modeling language, automatic differentiation and optimization solver.

for model predictive control in a centralized architecture [184,185]. For distributed control and optimization, open-source software packages are lacking.

5. Case Study on a Benchmark Process

In the works of our group, we have examined network decomposition and distributed optimization methods applied to the distributed MPC of a benchmark reactor–separator process [186,187]. This system can be regarded as a minimal process network for distributed control and used for a demonstration of concept. For more complex processes, such as those appearing in oil refineries, in-depth investigations need to be carried out and reported.

As illustrated in Fig. 10, the process comprises two reactors, where first-order reactions $A \rightarrow B \rightarrow C$ take place, and a separator from which the vapor phase is recycled and the liquid phase is withdrawn as product. The detailed dynamics and values of the parameters, steady states and inputs can be found in the aforementioned papers. The model contains 12 state variables, including holdup volumes (V_1, V_2, V_3), and temperatures (T_1, T_2, T_3) and concentrations ($x_{A1}, x_{A2}, x_{A3}, x_{B1}, x_{B2}, x_{B3}$), and 9 inputs, including flow rates ($F_{f1}, F_{f2}, F_1, F_2, F_3, F_R$) and heat exchange rates (Q_1, Q_2, Q_3). From the 12 states, 9 are considered as outputs (except for x_{A1}, x_{A2}, x_{A3}).

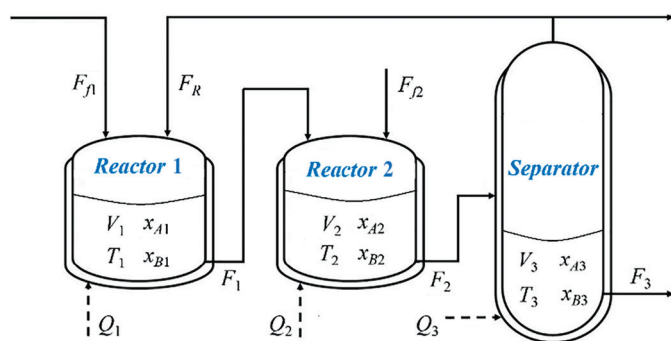


Fig. 10. Reactor–separator process.

5.1. Studies on network decomposition methods

For the reactor–separator process, we have considered the effect of network decomposition methods on the distributed model predictive control. Different network decompositions were generated based on respective decomposition methods, as summarized in Table 1. Specifically,

- Decomposition 1 is a centralized architecture with a single subsystem containing all inputs and outputs.
- Decomposition 2 is obtained by community detection based on relative time-averaged gain array (RTAGA) [113] that captures the finite-time interaction strengths between the variables. The RTAGA techniques tend to generate fine decompositions with a large number of subsystems.
- Decomposition 3 is obtained by community detection in a weighted directed graph representation that captures the short-time interaction strengths between the variables [112].
- Decomposition 4 is obtained by community detection in the unweighted directed graph representation that captures the structural interactions between the variables [111].
- Decomposition 5 is intuitively obtained based on the network topology (reactor 1 + reactor 2 + separator). The specific topology of the reactor–separator process makes it similar to decomposition 4.

Table 1

Decompositions obtained by different methods for the benchmark process

Decomp.	Subsyst.	Inputs	Outputs
1	All	All	All
2 [113]	I	F_{f1}, F_R	V_1, x_{B1}
	II	F_{f2}, F_1	V_2, x_{B2}
	III	F_2	x_{B3}
	IV	F_3	V_3
	V	Q_1	T_1
	VI	Q_2	T_2
	VII	Q_3	T_3
3 [112]	I	F_{f1}, Q_1	T_1, x_{B1}
	II	F_{f2}, Q_2, Q_3	T_2, T_3, x_{B2}
	III	F_1	V_1, V_2
	IV	F_2, F_3, F_R	V_3, x_{B3}
4 [111]	I	F_{f1}, F_R, Q_1	V_1, T_1, x_{B1}
	II	F_{f2}, F_1, Q_2	V_2, T_2, x_{B2}
	III	F_2, F_3, Q_3	V_3, T_3, x_{B3}
5	I	F_{f1}, F_1, Q_1	V_1, T_1, x_{B1}
	II	F_{f2}, F_2, Q_2	V_2, T_2, x_{B2}
6	III	F_3, F_R, Q_3	V_3, T_3, x_{B3}
	I	F_{f1}, F_{f2}, F_R	x_{B1}, x_{B2}, x_{B3}
	II	F_1, F_2, F_3	V_1, V_2, V_3
	III	Q_1, Q_2, Q_3	T_1, T_2, T_3

- Decomposition 6 is intuitively obtained considering mass and energy balances (feed and recycle flow rates affect concentrations; inventory flow rates affect holdup volumes, heat exchange rates affect temperature).

Distributed MPC was simulated for these decompositions for a tracking control task in 6 h [187], during which the setpoint changes every 2 h. The integrated squared error plus integrated control signals (ISE + ISC), namely the objective function in the optimal control problem of MPC, was used as a control performance index. A low performance index indicates good performance. By comparing the performance indices and computational times for the 6 decompositions, it was observed that

- Centralized MPC has a high computational burden compared to distributed MPC, and is hence not favorable.
- Among the community detection methods, the method of [111] (decomposition 4) results in control performance comparable to the centralized MPC (7.2×10^3 compared to 6.5×10^3), while largely reducing the computational time (2.6 min compared to 15.0 min). It also outcompetes the method of [112] (decomposition 3). The method of [113] (decomposition 2) has a deteriorated performance but a low computational cost due to a finer decomposition into 7 subsystems.
- The intuitive decomposition 6 is not favored, with a higher performance index and longer computational time than decomposition 4. Decomposition 5 has very close performance to decomposition 4, since the subsystem assignments of the variables are similar.

Therefore, we advocate community detection, especially the method of [111], as a beneficial tool to generate a high-quality decomposition for distributed control.

5.2. Studies on distributed optimization methods

We also studied distributed optimization methods for the reactor–separator benchmark process. In [187], the computational times of distributed MPC were compared using two different solvers — IPOPT and the traditional sequential quadratic programming (SQP). It was noted that IPOPT reduces the computational time approximately 60% compared to SQP for all the decompositions mentioned in the previous subsection.

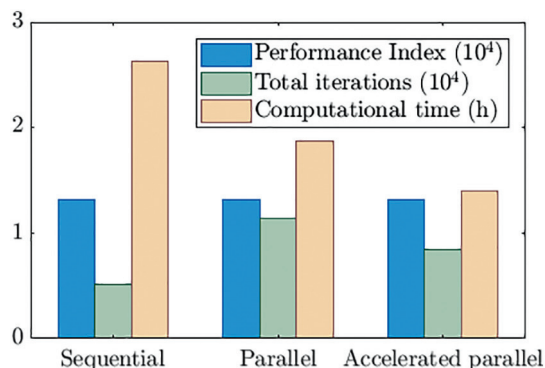


Fig. 11. Comparison of different ADMM algorithms.

When the agents only know the model of the corresponding subsystem instead of the whole plant, BCD cannot be used as a localized distributed optimization algorithm. In such a setting, we used the decomposition 4 in Table 1 and consider the tracking control of the benchmark process. 3 ADMM algorithms were investigated, namely sequential (Gauss–Seidel), parallel (Jacobi-proximal) and accelerated parallel (discounted Nesterov), depending on the way that the primal variables are updated [160]. Under the ADMM algorithms, the closed-loop system is stabilized in 5 h. Fig. 11 shows the comparison of 3 ADMM algorithms in terms of their performance indices, total ADMM iterations, and computational times (with an IPOPT solver).

- Among the ADMM algorithms, the accelerated parallel algorithm reduces the computational time up to 47% compared to the classical sequential algorithm. On the other hand, by using parallelization, the communication cost (number of iterations) inevitably increases.
- All ADMM algorithms lead to identical control trajectories (see [160]) with a stabilizing time of about 5 h and a performance index of 1.32×10^4 . In this sense, distributed algorithms of the ADMM type are more robust than BCD.
- Both the control performance and computational efficiency of ADMM are deteriorated compared to BCD. The underlying reason is that when the agents only know the models of the local subsystems, it takes more iterations to coordinate the agents in ADMM. Although in ADMM, each optimization step based on local prediction is simpler than a BCD optimization step, due to the small size of the plant, this advantage of ADMM is not sufficient to overcome the demand for a large number of iterations for coordination in ADMM.

Therefore, it appears that BCD algorithms perform significantly better than ADMM if the system model is moderately complex and unchanged, while ADMM algorithms may be more suitable when the system model is large or frequently changing that BCD optimization steps based on the entire system information is intractable or unsuitable.

6. Future Directions

In the previous sections, we have reviewed up-to-date methods of distributed control and optimization. For the future development and industrial application of distributed control and optimization methods, in our opinion, the following directions are of importance:

- The communication network-induced issues need to be addressed in a more unifying framework that incorporates nonlinear systems and optimization-based, distributed agent decisions in a closed loop.

- The network decomposition and distributed optimization algorithms should be investigated for larger industrial systems, including those with units governed by partial differential equations. Comparison of different decomposition algorithms and different distributed optimization algorithms should be reported. Model reduction techniques may be necessary to make the distributed optimization algorithm real-time implementable for plant-wide systems.
- Algorithms lie at the heart of distributed control and optimization. The nonconvex nature of the control and optimization problems in process systems tends to increase the computational burden, and makes it more difficult to accelerate the algorithms than for convex problems. Moreover, for nonconvex problems, we are often required to find a global optimum rather than a local one, which gives rise to the problem of distributed global optimization. To this end, augmented Lagrangian-based global optimization algorithms [188] combined with global solvers such as ANTIGONE [175] or BARON [174], offer potential for distributed implementation.
- The modeling, decomposition and distributed control and optimization framework should be integrated into efficient software packages, so that the users do not have to run through all the details of these methods, as long as the necessary settings and information are provided. We notice the efforts to incorporate graph topology into distributed optimization software [189,190]. It is also desirable to bridge this platform to existing commercial software for process design.
- Process monitoring, *i.e.* fault detection and diagnosis, is highly important to the process safety and product quality. Distributed process monitoring techniques have been proposed for both statistical [191,192] and model-based [193,194] approaches, where the monitors detect faults in the corresponding subsystems, and communicate to form the overall decision. For more advanced approaches *e.g.* using neural networks [195,196], distributed monitoring has not yet been addressed. Network decomposition methods for monitoring and their effects on the monitoring performance have only been studied recently [197].
- Data-driven control has been proposed as an alternative to the model-based control if an accurate control-oriented process model cannot be obtained [198]. Data-driven control methods are mostly based on data-driven model identification (see [199]), or reinforcement learning (see [200]). So far, these methods are apparently applicable only to small-scale systems that are not highly nonlinear. Distributed algorithms for data-driven control and optimization of process networks are apparently in need.

7. Conclusions

Large-scale, complex process networks are prevalent in chemical and other industries due to economics and sustainability considerations. Distributed control and optimization have been identified as a crucial decision-making paradigm for the operation of such complex networks, and studied extensively in the recent years. In this review paper, we first introduced the basic concepts of distributed control and optimization. We reviewed two most important aspects of distributed control and optimization, namely the methods of determining a network decomposition, and the methods of performing optimization based on a network decomposition. A case study was used to illustrate these methods. We also briefly proposed some potential future directions.

References

- [1] P. Daoutidis, M. Zachar, S.S. Jogwar, Sustainability and process control: A survey and perspective, *J. Process Control* 44 (2016) 184–206.
- [2] J.J. Sirola, T.F. Edgar, Process energy systems: Control, economic, and sustainability objectives, *Comput. Chem. Eng.* 47 (2012) 134–144.

- [3] S.S. Jogwar, A.I. Torres, P. Daoutidis, Networks with large solvent recycle: Dynamics, hierarchical control, and a biorefinery application, *AIChE J.* 58 (2012) 1764–1777.
- [4] M. Baldea, N.H. El-Farra, B.E. Ydstie, Dynamics and control of chemical process networks: Integrating physics, communication and computation, *Comput. Chem. Eng.* 51 (2013) 42–54.
- [5] N.K. Shah, M.G. Ierapetritou, Short-term scheduling of a large-scale oil-refinery operations: Incorporating logistics details, *AIChE J.* 57 (2011) 1570–1584.
- [6] M. Zachar, P. Daoutidis, Microgrid/macrogrid energy exchange: A novel market structure and stochastic scheduling, *IEEE Trans. Smart Grid* 8 (2017) 178–189.
- [7] M.J. Risbeck, C.T. Maravelias, J.B. Rawlings, R.D. Turney, A mixed-integer linear programming model for real-time cost optimization of building heating, ventilation, and air conditioning equipment, *Energy Build.* 142 (2017) 220–235.
- [8] D.J. Garcia, F. You, Supply chain design and optimization: Challenges and opportunities, *Comput. Chem. Eng.* 81 (2015) 153–170.
- [9] A.P. Barbosa-Póvoa, Progresses and challenges in process industry supply chains optimization, *Curr. Opin. Chem. Eng.* 1 (2012) 446–452.
- [10] M. Baldea, P. Daoutidis, Dynamics and Nonlinear Control of Integrated Process Systems, Cambridge University Press, 2012.
- [11] D. Yue, F. You, Optimal supply chain design and operations under multi-scale uncertainties: Nested stochastic robust optimization modeling framework and solution algorithm, *AIChE J.* 62 (2016) 3041–3055.
- [12] L. Luo, F. Cheng, T. Qiu, J. Zhao, Refined convergent cross-mapping for disturbance propagation analysis of chemical processes, *Comput. Chem. Eng.* 106 (2017) 1–16.
- [13] S. Rivero, M. Farina, G. Ferrari-Trecate, Plug-and-play state estimation and application to distributed output-feedback model predictive control, *Eur. J. Control* 25 (2015) 17–26.
- [14] D. Chilin, J. Liu, D. Muñoz de la Peña, P.D. Christofides, J.F. Davis, Detection, isolation and handling of actuator faults in distributed model predictive control systems, *J. Process Control* 20 (2010) 1059–1075.
- [15] S. Rivero, F. Boem, G. Ferrari-Trecate, T. Parisini, Plug-and-play fault detection and control-reconfiguration for a class of nonlinear large-scale constrained systems, *IEEE Trans. Autom. Control* 61 (2016) 3963–3978.
- [16] D.E. Seborg, D.A. Mellichamp, T.F. Edgar, F.J. Doyle III, Process Dynamics and Control, John Wiley & Sons, 2010.
- [17] C.T. Maravelias, C. Sung, Integration of production planning and scheduling: Overview, challenges and opportunities, *Comput. Chem. Eng.* 33 (2009) 1919–1930.
- [18] Y. Chu, F. You, Model-based integration of control and operations: Overview, challenges, advances, and opportunities, *Comput. Chem. Eng.* 83 (2015) 2–20.
- [19] L.S. Dias, M.G. Ierapetritou, From process control to supply chain management: An overview of integrated decision making strategies, *Comput. Chem. Eng.* 106 (2017) 826–835.
- [20] P. Daoutidis, J.H. Lee, I. Harjunkski, S. Skogestad, M. Baldea, C. Georgakis, Integrating operations and control: A perspective and roadmap for future research, *Comput. Chem. Eng.* 115 (2018) 179–184.
- [21] I.E. Grossmann, Enterprise-wide optimization: A new frontier in process systems engineering, *AIChE J.* 51 (2005) 1846–1857.
- [22] I.E. Grossmann, Advances in mathematical programming models for enterprise-wide optimization, *Comput. Chem. Eng.* 47 (2012) 2–18.
- [23] C.T. Maravelias, General framework and modeling approach classification for chemical production scheduling, *AIChE J.* 58 (2012) 1812–1828.
- [24] I. Harjunkski, C.T. Maravelias, P. Bongers, P.M. Castro, S. Engell, I.E. Grossmann, J. Hooker, C. Méndez, G. Sand, J. Wassick, Scope for industrial applications of production scheduling models and solution methods, *Comput. Chem. Eng.* 62 (2014) 161–193.
- [25] P.M. Castro, I.E. Grossmann, Q. Zhang, Expanding scope and computational challenges in process scheduling, *Comput. Chem. Eng.* 114 (2018) 14–42.
- [26] J.B. Rawlings, D.Q. Mayne, Model Predictive Control: Theory and Design, Nob Hill Publishing, 2009.
- [27] D.Q. Mayne, Model predictive control: Recent developments and future promise, *Automatica* 50 (2014) 2967–2986.
- [28] J.B. Rawlings, D. Angeli, C.N. Bates, Fundamentals of economic model predictive control, in: Proc. 51st Conf. Decis. Control (CDC), IEEE, 2012, pp. 3851–3861.
- [29] M. Ellis, J. Liu, P.D. Christofides, Economic Model Predictive Control, Springer, 2016.
- [30] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice Hall, 1989.
- [31] C. Langbort, R.S. Chandra, R. D'Andrea, Distributed control design for systems interconnected over an arbitrary graph, *IEEE Trans. Autom. Control* 49 (2004) 1502–1519.
- [32] F. Lin, M. Fardad, M.R. Jovanović, Design of optimal sparse feedback gains via the alternating direction method of multipliers, *IEEE Trans. Autom. Control* 58 (2013) 2426–2431.
- [33] D. Hioe, J. Bao, B.E. Ydstie, Dissipativity analysis for networks of process systems, *Comput. Chem. Eng.* 50 (2013) 207–219.
- [34] M.J. Tippet, J. Bao, Dissipativity based distributed control synthesis, *J. Process Control* 23 (2013) 755–766.
- [35] R. Scattolini, Architectures for distributed and hierarchical model predictive control – A review, *J. Process Control* 19 (2009) 723–731.
- [36] P.D. Christofides, R. Scattolini, D. Muñoz de la Peña, J. Liu, Distributed model predictive control: A tutorial review and future research directions, *Comput. Chem. Eng.* 51 (2013) 21–41.
- [37] R.R. Negenborn, J.M. Maestre, Distributed model predictive control: An overview and roadmap of future research opportunities, *IEEE Control. Syst.* 34 (2014) 87–97.
- [38] N. Gatsis, G.B. Giannakis, Residential load control: Distributed scheduling and convergence with lost AMI messages, *IEEE Trans. Smart Grid* 3 (2012) 770–786.
- [39] S.-C. Tsai, Y.-H. Tseng, T.-H. Chang, Communication-efficient distributed demand response: A randomized ADMM approach, *IEEE Trans. Smart Grid* 8 (2017) 1085–1095.
- [40] A.T. Georgiou, K.R. Sheth, Plant-wide dynamic economic optimization: Key challenges and new opportunities, in: Foundations of Computer Aided Process Operations / Chemical Process Control (FOCAPO/CPC 2017), 2017. <http://focapocpc.org/pdf/Georgiou.pdf>.
- [41] S. Harwood, S. Kameswaran, T.A. Badgwell, Analysis of the alternating direction method of multipliers for the optimization of distributed nonconvex systems, in: 2017 AIChE Annual Meeting, 2017. <https://www.aiche.org/conferences/aiche-annualmeeting/2017/proceeding/paper/522f-analysis-alternatingdirection-method-multipliers-optimization-distributed-nonconvex-systems>.
- [42] R.A. Gupta, M.-Y. Chow, Networked control system: Overview and research trends, *IEEE Trans. Ind. Electron.* 57 (2010) 2527–2535.
- [43] X. Ge, F. Yang, Q.-L. Han, Distributed networked control systems: A brief overview, *Inf. Sci.* 380 (2017) 117–131.
- [44] L. Zhang, H. Gao, O. Kaynak, Network-induced constraints in networked control systems – A survey, *IEEE Trans. Ind. Inform.* 9 (2013) 403–416.
- [45] R. Zhang, J. Kwok, Asynchronous distributed ADMM for consensus optimization, *Proc. 31st Int. Conf. Mach. Learn. (ICML)* (2014) 1701–1709.
- [46] T.-H. Chang, M. Hong, W.-C. Liao, X. Wang, Asynchronous distributed ADMM for large-scale optimization – Part I: Algorithm and convergence analysis, *IEEE Trans. Signal Process.* 64 (2016) 3118–3130.
- [47] M. Schmidt, N.L. Roux, F.R. Bach, Convergence rates of inexact proximal-gradient methods for convex optimization, *Proc. 24th Int. Conf. Neural Inf. Process. Syst. (NIPS'11)* (2011) 1458–1466.
- [48] Y. Pu, M.N. Zeilinger, C.N. Jones, Inexact fast alternating minimization algorithm for distributed model predictive control, in: Proc. 53rd Conf. Decis. Control (CDC), IEEE, 2014, pp. 5915–5921.
- [49] Y. Pu, M.N. Zeilinger, C.N. Jones, Quantization design for distributed optimization, *IEEE Trans. Autom. Control* 62 (2017) 2107–2120.
- [50] M. Zhong, C.G. Cassandras, Asynchronous distributed optimization with event-driven communication, *IEEE Trans. Autom. Control* 55 (2010) 2735–2750.
- [51] M. Meinel, M. Ulbrich, S. Albrecht, A class of distributed optimization methods with event-triggered communication, *Comput. Optim. Appl.* 57 (2014) 517–553.
- [52] P. Wan, M.D. Lemmon, Optimal power flow in microgrids using event-triggered optimization, *Proc. 2010 Amer. Control Conf. (ACC)*, IEEE (2010) 2521–2526.
- [53] C. Li, X. Yu, W. Yu, T. Huang, Z.-W. Liu, Distributed event-triggered scheme for economic dispatch in smart grids, *IEEE Trans. Ind. Inform.* 12 (2016) 1775–1785.
- [54] T. Yang, Trading computation for communication: Distributed stochastic dual coordinate ascent, *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS'13)* (2013) 629–637.
- [55] M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, M.I. Jordan, Communication-efficient distributed dual coordinate ascent, *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS'14)* (2014) 3068–3076.
- [56] O. Shamir, N. Srebro, T. Zhang, Communication-efficient distributed optimization using an approximate Newton-type method, *Proc. 31st Int. Conf. Mach. Learn. (ICML)* (2014) 1000–1008.
- [57] S.J. Reddi, J. Konečný, P. Richtárik, B. Póczos, A. Smola, AIDE: Fast and communication efficient distributed optimization arXiv preprint arXiv:1608.06879, 2016.
- [58] M.B.G. Cloosterman, N. van de Wouw, W. Heemels, H. Nijmeijer, Stability of networked control systems with uncertain time-varying delays, *IEEE Trans. Autom. Control* 54 (2009) 1575–1580.
- [59] H. Lin, P.J. Antsaklis, Stability and stabilizability of switched linear systems: A survey of recent results, *IEEE Trans. Autom. Control* 54 (2009) 308–322.
- [60] J. Liu, D. Muñoz de la Peña, P.D. Christofides, Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements, *Automatica* 46 (2010) 52–61.
- [61] J. Liu, X. Chen, D. Muñoz de la Peña, P.D. Christofides, Iterative distributed model predictive control of nonlinear systems: Handling asynchronous, delayed measurements, *IEEE Trans. Autom. Control* 57 (2012) 528–534.
- [62] H. Li, Y. Shi, Distributed receding horizon control of large-scale nonlinear systems: Handling communication delays and disturbances, *Automatica* 50 (2014) 1264–1271.
- [63] D. Yue, Q.-L. Han, C. Peng, State feedback controller design of networked control systems, in: Proc. 2004 Int. Conf. Control Appl., IEEE, 2004, pp. 242–247.
- [64] D.E. Quevedo, D. Nešić, Input-to-state stability of packetized predictive control over unreliable networks affected by packet dropouts, *IEEE Trans. Autom. Control* 56 (2011) 370–375.
- [65] Y.-B. Zhao, J. Kim, G.-P. Liu, D. Rees, Compensation and stochastic modeling of discrete-time networked control systems with data packet disorder, *Int. J. Control Autom. Syst.* 10 (2012) 1055–1063.
- [66] M. Heidarinejad, J. Liu, D. Muñoz de la Peña, J.F. Davis, P.D. Christofides, Handling communication disruptions in distributed model predictive control, *J. Process Control* 21 (2011) 173–181.

- [67] H. Li, M.-Y. Chow, Z. Sun, Optimal stabilizing gain selection for networked control systems with time delays and packet losses, *IEEE Trans. Control Syst. Technol.* 17 (2009) 1154–1162.
- [68] L.A. Montestruque, P. Antsaklis, Stability of model-based networked control systems with time-varying transmission times, *IEEE Trans. Autom. Control* 49 (2004) 1562–1572.
- [69] I.G. Polushin, P.X. Liu, C.-H. Lung, On the model-based approach to nonlinear networked control systems, *Automatica* 44 (2008) 2409–2414.
- [70] M. Tabbara, D. Nešić, A.R. Teel, Stability of wireless and wireline networked control systems, *IEEE Trans. Autom. Control* 52 (2007) 1615–1630.
- [71] W.M.H. Heemels, A.R. Teel, N. Van de Wouw, D. Nešić, Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance, *IEEE Trans. Autom. Control* 55 (2010) 1781–1796.
- [72] E. Fridman, M. Dambrine, Control under quantization, saturation and delay: An LMI approach, *Automatica* 45 (2009) 2258–2264.
- [73] Y. Ishido, K. Takaba, D.E. Quevedo, Stability analysis of networked control systems subject to packet-dropouts and finite level quantization, *Syst. Control Lett.* 60 (2011) 325–332.
- [74] R. Postoyan, N. Van de Wouw, D. Nešić, W.M.H. Heemels, Tracking control for nonlinear networked control systems, *IEEE Trans. Autom. Control* 59 (2014) 1539–1554.
- [75] F. Forni, S. Galeani, D. Nešić, L. Zaccarian, Event-triggered transmission for linear control over communication channels, *Automatica* 50 (2014) 490–498.
- [76] R. Postoyan, P. Tabuada, D. Nešić, A. Anta, A framework for the event-triggered stabilization of nonlinear systems, *IEEE Trans. Autom. Control* 60 (2015) 982–996.
- [77] A. Girard, Dynamic triggering mechanisms for event-triggered control, *IEEE Trans. Autom. Control* 60 (2015) 1992–1997.
- [78] H. Li, Y. Shi, Event-triggered robust model predictive control of continuous-time nonlinear systems, *Automatica* 50 (2014) 1507–1513.
- [79] F.D. Brunner, M. Heemels, F. Allgöwer, Robust self-triggered MPC for constrained linear systems: A tube-based approach, *Automatica* 72 (2016) 73–83.
- [80] P. Daoutidis, W. Tang, S.S. Jogwar, Decomposing complex plants for distributed control: Perspectives from network theory, *Comput. Chem. Eng.* 114 (2018) 43–51.
- [81] I.S. Duff, A survey of sparse matrix research, *Proc. IEEE* 65 (1977) 500–535.
- [82] D.B. West, Introduction to Graph Theory, Prentice Hall, 2001.
- [83] A. Zečević, D.D. Šiljak, Control of Complex Systems: Structural Constraints and Uncertainty, Springer, 2010.
- [84] A. Michel, R. Miller, W. Tang, Lyapunov stability of interconnected systems: Decomposition into strongly connected subsystems, *IEEE Trans. Circ. Syst.* 25 (1978) 799–809.
- [85] V. Pichai, M.E. Sezer, D.D. Šiljak, A graph-theoretic algorithm for hierarchical decomposition of dynamic systems with applications to estimation and control, *IEEE Trans. Syst. Man Cybern.* 13 (1983) 197–207.
- [86] D.D. Šiljak, Large-scale Dynamic Systems: Stability and Structure, North Holland, 1978.
- [87] C.-E. Bichot, P. Siarry, Graph Partitioning, Wiley, 2011.
- [88] K. Hangos, Zs. Tuza, Optimal control structure selection for process systems, *Comput. Chem. Eng.* 25 (2001) 1521–1536.
- [89] J. Anderson, A. Papachristodoulou, A decomposition technique for nonlinear dynamical system analysis, *IEEE Trans. Autom. Control* 57 (2012) 1516–1521.
- [90] C. Ocampo-Martínez, S. Bovo, V. Puig, Partitioning approach oriented to the decentralised predictive control of large-scale systems, *J. Process Control* 21 (2011) 775–786.
- [91] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, C. Schulz, Recent advances in graph partitioning, in: L. Kliemann, P. Sanders (Eds.), Algorithm Engineering: Selected Results and Surveys, Springer, 2016, pp. 117–158.
- [92] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: A review, *ACM Comput. Surv. (CSUR)* 31 (1999) 264–323.
- [93] R. Xu, D. Wunsch, Survey of clustering algorithms, *IEEE Trans. Neural Netw.* 16 (2005) 645–678.
- [94] F. Murtagh, P. Contreras, Algorithms for hierarchical clustering: An overview, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2 (2012) 86–97.
- [95] P. Daoutidis, C. Kravaris, Structural evaluation of control configurations for multivariable nonlinear processes, *Chem. Eng. Sci.* 47 (1992) 1091–1107.
- [96] S. Heo, W.A. Marwin, P. Daoutidis, Automated synthesis of control configurations for process networks based on structural coupling, *Chem. Eng. Sci.* 136 (2015) 76–87.
- [97] L. Kang, W. Tang, Y. Liu, P. Daoutidis, Control configuration synthesis using agglomerative hierarchical clustering: A graph-theoretic approach, *J. Process Control* 46 (2016) 43–54.
- [98] M. Moharir, L. Kang, P. Daoutidis, A. Almansoori, Graph representation and decomposition of ODE/hyperbolic PDE systems, *Comput. Chem. Eng.* 106 (2017) 532–543.
- [99] S. Heo, P. Daoutidis, Control-relevant decomposition of process networks via optimization-based hierarchical clustering, *AIChE J.* 62 (2016) 3177–3188.
- [100] A.-L. Barabási, Network Science, Cambridge University Press, 2016.
- [101] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, *Proc. Natl. Acad. Sci.* 99 (2002) 7821–7826.
- [102] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, *J. Stat. Mech. Theory Exp.* 2008 (2008) P10008.
- [103] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, *Phys. Rev. E* 69 (2004) 026113.
- [104] M.E.J. Newman, Modularity and community structure in networks, *Proc. Natl. Acad. Sci.* 103 (2006) 8577–8582.
- [105] S. Fortunato, D. Hric, Community detection in networks: A user guide, *Phys. Rep.* 659 (2016) 1–44.
- [106] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, *Phys. Rev. E* 76 (2007) 036106.
- [107] J. Xie, B.K. Szymanski, X. Liu, SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, *Proc. 11th Int. Conf. Data Min. Data Min. Workshops (ICDMW)*, IEEE (2011) 344–349.
- [108] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, *Proc. Natl. Acad. Sci. U. S. A.* 105 (2008) 1118–1123.
- [109] B. Karrer, M.E.J. Newman, Stochastic blockmodels and community structure in networks, *Phys. Rev. E* 83 (2011) 016107.
- [110] T.P. Peixoto, Nonparametric Bayesian inference of the microcanonical stochastic block model, *Phys. Rev. E* 95 (2017) 012317.
- [111] S.S. Jogwar, P. Daoutidis, Community-based synthesis of distributed control architectures for integrated process networks, *Chem. Eng. Sci.* 172 (2017) 434–443.
- [112] W. Tang, P. Daoutidis, Network decomposition for distributed control through community detection in input–output bipartite graphs, *J. Process Control* 64 (2018) 7–14.
- [113] W. Tang, D.B. Pourkargar, P. Daoutidis, Relative time-averaged gain array (RTAGA) for distributed control-oriented network decomposition, *AIChE J.* 64 (2018) 1682–1690.
- [114] M. Ellis, P.D. Christofides, Selection of control configurations for economic model predictive control systems, *AIChE J.* 60 (2014) 3230–3242.
- [115] R. Chen, K. Dave, T.J. McAvoy, M. Luyben, A nonlinear dynamic model of a vinyl acetate process, *Ind. Eng. Chem. Res.* 42 (2003) 4478–4487.
- [116] E.A. del Río-Chanona, F. Fiorelli, V.S. Vassiliadis, Automated structure detection for distributed process optimization, *Comput. Chem. Eng.* 89 (2016) 135–148.
- [117] W. Tang, A. Allman, D.B. Pourkargar, P. Daoutidis, Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection, *Comput. Chem. Eng.* 111 (2018) 43–54.
- [118] A. Allman, W. Tang, P. Daoutidis, Towards a generic algorithm for identifying high-quality decompositions of optimization problems, *Comput. Aided Chem. Eng.* 44 (2018) 943–948.
- [119] S.J. Wright, Coordinate descent algorithms, *Math. Program.* 151 (2015) 3–34.
- [120] J. Liu, D. Muñoz de la Peña, P.D. Christofides, Distributed model predictive control of nonlinear process systems, *AIChE J.* 55 (2009) 1171–1184.
- [121] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, G. Pannocchia, Cooperative distributed model predictive control, *Syst. Control Lett.* 59 (2010) 460–469.
- [122] M. Heidarinejad, J. Liu, P.D. Christofides, Distributed model predictive control of switched nonlinear systems with scheduled mode transitions, *AIChE J.* 59 (2013) 860–871.
- [123] M. Heidarinejad, J. Liu, P.D. Christofides, Economic model predictive control of nonlinear process systems using Lyapunov techniques, *AIChE J.* 58 (2012) 855–870.
- [124] F. Albalawi, H. Durand, P.D. Christofides, Distributed economic model predictive control for operational safety of nonlinear processes, *AIChE J.* 63 (2017) 3404–3418.
- [125] Yu. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems, *SIAM J. Optim.* 22 (2012) 341–362.
- [126] P. Richtárik, M. Takáč, Parallel coordinate descent methods for big data optimization, *Math. Program.* 156 (2016) 433–484.
- [127] A. Beck, L. Tetruashvili, On the convergence of block coordinate descent type methods, *SIAM J. Optim.* 23 (2013) 2037–2060.
- [128] Z. Lu, L. Xiao, On the complexity analysis of randomized block-coordinate descent methods, *Math. Program.* 152 (2015) 615–642.
- [129] M. Hong, X. Wang, M. Razaviyayn, Z.-Q. Luo, Iterative complexity analysis of block coordinate descent methods, *Math. Program.* 163 (2017) 85–114.
- [130] Yu. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, *Sov. Math. Dokl.* 27 (1983) 372–376.
- [131] O. Fercoq, P. Richtárik, Accelerated, parallel, and proximal coordinate descent, *SIAM J. Optim.* 25 (2015) 1997–2023.
- [132] Z. Allen-Zhu, Z. Qu, P. Richtárik, Y. Yuan, Even faster accelerated coordinate descent using non-uniform sampling, *Proc. 33rd Int. Conf. Mach. Learn. (ICML)* (2016) 1110–1119.
- [133] Y. Xu, W. Yin, A globally convergent algorithm for nonconvex optimization based on block coordinate update, *J. Sci. Comput.* 72 (2017) 700–734.
- [134] J. Liu, X. Chen, D. Muñoz de la Peña, P.D. Christofides, Sequential and iterative architectures for distributed model predictive control of nonlinear process systems, *AIChE J.* 56 (2010) 2137–2149.
- [135] M. Guignard, S. Kim, Lagrangean decomposition: A model yielding stronger Lagrangean bounds, *Math. Program.* 39 (1987) 215–228.
- [136] M.L. Fisher, An applications-oriented guide to Lagrangian relaxation, *Interfaces* 15 (1985) 10–21.
- [137] H. Schramm, J. Zowe, A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results, *SIAM J. Optim.* 2 (1992) 121–152.
- [138] X. Zhao, P.B. Luh, J. Wang, Surrogate gradient algorithm for Lagrangian relaxation, *J. Optim. Theory Appl.* 100 (1999) 699–712.
- [139] D.P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, Academic Press, 1982.

- [140] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (2011) 1–122.
- [141] F. Oliveira, V. Gupta, S. Hamacher, I.E. Grossmann, A Lagrangean decomposition approach for oil supply chain investment planning under uncertainty with risk considerations, *Comput. Chem. Eng.* 50 (2013) 184–195.
- [142] B.R. Knudsen, I.E. Grossmann, B. Foss, A.R. Conn, Lagrangian relaxation based decomposition for well scheduling in shale-gas systems, *Comput. Chem. Eng.* 63 (2014) 234–249.
- [143] Y. Wakasa, M. Arakawa, K. Tanaka, T. Akashi, Decentralized model predictive control via dual decomposition, in: Proc. 47th Conf. Decis. Control (CDC), IEEE, 2008, pp. 381–386.
- [144] P. Giselsson, M.D. Doan, T. Keviczky, B. De Schutter, A. Rantzer, Accelerated gradient methods and dual decomposition in distributed model predictive control, *Automatica* 49 (2013) 829–833.
- [145] A. Grancharova, T.A. Johansen, Distributed MPC of interconnected nonlinear systems by dynamic dual decomposition, in: J.M. Maestre, R.R. Negenborn (Eds.), Distributed Model Predictive Control Made Easy, Springer, 2014, pp. 293–308.
- [146] F. Farokhi, I. Shames, K.H. Johansson, Distributed MPC via dual decomposition and alternative direction method of multipliers, in: J.M. Maestre, R.R. Negenborn (Eds.), Distributed Model Predictive Control Made Easy, Springer, 2014, pp. 115–131.
- [147] J.F.C. Mota, J.M.F. Xavier, P.M.Q. Aguiar, M. Püschel, Distributed optimization with local domains: Applications in MPC and network flows, *IEEE Trans. Autom. Control* 60 (2015) 2004–2009.
- [148] V. Kekatos, G.B. Giannakis, Distributed robust power system state estimation, *IEEE Trans. Power Syst.* 28 (2013) 1617–1626.
- [149] T. Erseghe, Distributed optimal power flow using ADMM, *IEEE Trans. Power Syst.* 29 (2014) 2370–2380.
- [150] W. Deng, M.-J. Lai, Z. Peng, W. Yin, Parallel multi-block ADMM with $o(1/k)$ convergence, *J. Sci. Comput.* 71 (2017) 712–736.
- [151] R. Sun, Z.-Q. Luo, Y. Ye, On the expected convergence of randomly permuted ADMM arXiv preprint arXiv:1503.06387, 2015.
- [152] T. Lin, S. Ma, S. Zhang, On the global linear convergence of the ADMM with multiblock variables, *SIAM J. Optim.* 25 (2015) 1478–1497.
- [153] M. Hong, Z.-Q. Luo, On the linear convergence of the alternating direction method of multipliers, *Math. Program.* 162 (2017) 165–199.
- [154] T. Goldstein, B. O'Donoghue, S. Setzer, R. Baraniuk, Fast alternating direction optimization methods, *SIAM J. Imaging Sci.* 7 (2014) 1588–1623.
- [155] Y. Ouyang, Y. Chen, G. Lan, E. Pasiliao Jr., An accelerated linearized alternating direction method of multipliers, *SIAM J. Imaging Sci.* 8 (2015) 644–681.
- [156] E. Ghadimi, A. Teixeira, I. Shames, M. Johansson, Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems, *IEEE Trans. Autom. Control* 60 (2015) 644–658.
- [157] Z. Xu, M. Figueiredo, T. Goldstein, Adaptive ADMM with spectral penalty parameter selection, *Proc. 20th Int. Conf. Artif. Intell. Stat.* (2016) 718–727.
- [158] Y. Wang, W. Yin, J. Zeng, Global convergence of ADMM in nonconvex nonsmooth optimization arXiv preprint arXiv:1511.06324, 2015.
- [159] M. Hong, Z.-Q. Luo, M. Razaviyayn, Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems, *SIAM J. Optim.* 26 (2016) 337–364.
- [160] W. Tang, P. Daoutidis, Accelerated parallel alternating method of multipliers (ADMM) for distributed optimization, 2018 *AIChE Annual Meeting* (2018). <https://aiche.confex.com/aiche/2018/meetingapp.cgi/Paper/528146>.
- [161] A.V. Rao, D.A. Benson, C. Darby, M.A. Patterson, C. Francolin, I. Sanders, G.T. Huntington, Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method, *ACM Trans. Math. Softw. (TOMS)* 37 (2010) 22.
- [162] M.A. Patterson, A.V. Rao, GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming, *ACM Trans. Math. Softw. (TOMS)* 41 (1) (2014). <https://doi.org/10.1145/2558904>.
- [163] S. Velez, C.T. Maravelias, Reformulations and branching methods for mixed-integer programming chemical production scheduling models, *Ind. Eng. Chem. Res.* 52 (2013) 3832–3841.
- [164] H. Lee, C.T. Maravelias, Combining the advantages of discrete- and continuous-time scheduling models: Part 1. Framework and mathematical formulations, *Comput. Chem. Eng.* 116 (2018) 176–190.
- [165] J. Du, J. Park, I. Harjunkoski, M. Baldea, A time scale-bridging approach for integrating production scheduling and process control, *Comput. Chem. Eng.* 79 (2015) 59–69.
- [166] J. Zhuge, M.G. Ierapetritou, An integrated framework for scheduling and control using fast model predictive control, *AIChE J.* 61 (2015) 3304–3319.
- [167] M.T. Kelley, R.C. Pattison, R. Baldick, M. Baldea, An efficient MILP framework for integrating nonlinear process dynamics and control in optimal production scheduling calculations, *Comput. Chem. Eng.* 110 (2018) 35–52.
- [168] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Math. Program.* 106 (2006) 25–57.
- [169] R.H. Byrd, J. Nocedal, R.A. Waltz, KNITRO: An integrated package for nonlinear optimization, in: G. Di Pillo, M. Roma (Eds.), Large-scale Nonlinear Optimization, Springer, 2006, pp. 35–59.
- [170] L.T. Biegler, V.M. Zavala, Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization, *Comput. Chem. Eng.* 33 (2009) 575–582.
- [171] H. Pirnay, R. López-Negrete, L.T. Biegler, Optimal sensitivity based on IPOPT, *Math. Program. Comput.* 4 (2012) 307–331.
- [172] J. Kang, Y. Cao, D.P. Word, C.D. Laird, An interior-point method for efficient solution of block-structured NLP problems using an implicit Schur-complement decomposition, *Comput. Chem. Eng.* 71 (2014) 563–573.
- [173] W. Wan, J.P. Eason, B. Nicholson, L.T. Biegler, Parallel cyclic reduction decomposition for dynamic optimization problems, *Comput. Chem. Eng.* 120 (2019) 54–69.
- [174] N.V. Sahinidis, BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs, User's Manual, 2017.
- [175] R. Misener, C.A. Floudas, ANTIGONE: Algorithms for continuous/integer global optimization of nonlinear equations, *J. Glob. Optim.* 59 (2014) 503–526.
- [176] Y. Lin, L. Schrage, The global solver in the LINDO API, *Optim. Methods Softw.* 24 (2009) 657–668.
- [177] S. Vigerske, A. Gleixner, SCIP: Global optimization of mixed integer nonlinear programs in a branch-and-cut framework, *Optim. Methods Softw.* 33 (2018) 563–593.
- [178] P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wächter, Branching and bounds tightening techniques for non-convex MINLP, *Optim. Methods Softw.* 24 (2009) 597–634.
- [179] W.E. Hart, C.D. Laird, J.-P. Watson, D.L. Woodruff, G.A. Hackebeil, B.L. Nicholson, J.D. Sirola, Pyomo — Optimization Modeling in Python, 2nd edition., Springer, 2017.
- [180] I. Dunning, J. Huchette, M. Lubin, JuMP: A modeling language for mathematical optimization, *SIAM Rev.* 59 (2017) 295–320.
- [181] B. Nicholson, J.D. Sirola, J.-P. Watson, V.M. Zavala, L.T. Biegler, pyomo.dae: A modeling and automatic discretization framework for optimization with differential and algebraic equations, *Math. Program. Comput.* (2016) 1–37.
- [182] S.A. Forth, An efficient overloaded implementation of forward mode automatic differentiation in MATLAB, *ACM Trans. Math. Softw. (TOMS)* 32 (2006) 195–222.
- [183] J. Andersson, J. Åkesson, M. Diehl, CasADi: A symbolic package for automatic differentiation and optimal control, in: S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther (Eds.), Recent Advances in Algorithmic Differentiation, Springer, 2012, pp. 297–307.
- [184] J. Andersson, J. Åkesson, M. Diehl, Dynamic optimization with CasADi, in: Proc. 51th Conf. Decis. Control (CDC), IEEE, 2012, pp. 681–686.
- [185] M.J. Risbeck, J.B. Rawlings, MPCtools: Nonlinear model predictive control tools for CasADi (python interface) <https://bitbucket.org/rawlings-group/mpc-tools-casadi>, 2015.
- [186] D.B. Pourkargar, A. Almansoori, P. Daoutidis, Impact of decomposition on distributed model predictive control: A process network case study, *Ind. Eng. Chem. Res.* 56 (2017) 9606–9616.
- [187] D.B. Pourkargar, A. Almansoori, P. Daoutidis, Comprehensive study of decomposition effects on distributed output tracking of an integrated process over a wide operating range, *Chem. Eng. Res. Des.* 134 (2018) 553–563.
- [188] E.G. Birgin, J.M. Martínez, Practical Augmented Lagrangian Methods for Constrained Optimization, SIAM, 2014.
- [189] J. Jalving, V.M. Zavala, Y. Cao, A graph-based modeling and optimization framework for complex systems, 2017 *AIChE Annual Meeting* (2017). <https://www.aiche.org/conferences/aicheannual-meeting/2017/proceeding-paper/328g-graph-based-modeling-and-optimization-framework-complex-systems>.
- [190] J. Jalving, V.M. Zavala, An optimization-based state estimation framework for large-scale natural gas networks, *Ind. Eng. Chem. Res.* 57 (2018) 5966–5979.
- [191] Z. Ge, Z. Song, Distributed PCA model for plant-wide process monitoring, *Ind. Eng. Chem. Res.* 52 (2013) 1947–1957.
- [192] Q. Jiang, B. Huang, Distributed monitoring for large-scale processes based on multivariate statistical analysis and Bayesian method, *J. Process Control* 46 (2016) 75–83.
- [193] F. Boem, R.M.G. Ferrari, C. Keliris, T. Parisini, M.M. Polycarpou, A distributed networked approach for fault detection of large-scale systems, *IEEE Trans. Autom. Control* 62 (2017) 18–33.
- [194] H. Shahnazari, P. Mhaskar, Distributed fault diagnosis for networked nonlinear uncertain systems, *Comput. Chem. Eng.* 115 (2018) 22–33.
- [195] Z. Zhang, J. Zhao, A deep belief network based fault diagnosis model for complex chemical processes, *Comput. Chem. Eng.* 107 (2017) 395–407.
- [196] H. Wu, J. Zhao, Deep convolutional neural network model based chemical process fault diagnosis, *Comput. Chem. Eng.* 115 (2018) 185–197.
- [197] S. Khatib, P. Daoutidis, A. Almansoori, System decomposition for distributed multivariate statistical process monitoring by performance driven agglomerative clustering, *Ind. Eng. Chem. Res.* 57 (2018) 8283–8298.
- [198] J.H. Lee, W. Wong, Approximate dynamic programming approach for process control, *J. Process Control* 20 (2010) 1038–1048.
- [199] C. Novara, S. Formentin, Data-driven inversion-based control of nonlinear systems with guaranteed closed-loop stability, *IEEE Trans. Autom. Control* 63 (2018) 1147–1154.
- [200] F.L. Lewis, D. Liu, Reinforcement Learning and Approximate Dynamic Programming for Feedback Control, John Wiley & Sons, 2013.